Machine learning for modeling and interpreting geophysical borehole measurements

David Pardo[1, 2, 3], M. Mahdi Abedi[2], Mostafa Shahriari[4], Ali Hashemian[2], Jon A. Rivera[1], Ángel J. Omella[1]

[1]University of the Basque Country (UPV/EHU), Bilbao, Spain

[2]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain

[3]Ikerbasque, Bilbao, Spain

[4]Software Competence Center Hagenberg GmbH, Hagenberg, Austria

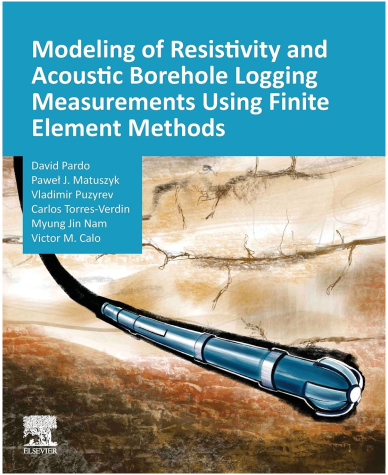
November 2nd, 2022
2nd Geosteering and Formation Evaluation Workshop
Stavanger, Norway

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

basque center for applied mathematics

Traditional Numerical Methods in Geophysics



Published in 2021

- Finite Element methods
- Finite Difference methods
- Finite Volume methods
- Integral methods
- Semi-analytical methods

Traditional Numerical Methods

Limitations:

Finite Element/Difference methods:

- Mesh dependent
- Fine grids for better accuracy $\Rightarrow$ High computational cost

Integral methods:

- Design of fast and robust integration techniques
- Dense matrices $\Rightarrow$ High computational cost

Why use Deep Learning for Geophysics?

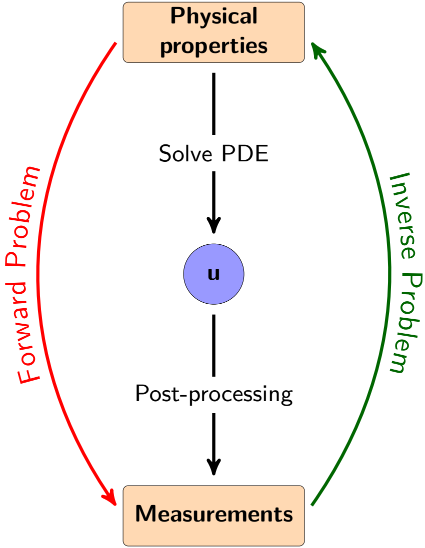
Deep Learning Methods

Advantages:

- Affordable computational cost (*High offline, low online*)
- Easily parallelizable implementation
- Great approximation capabilities
- Exploitable big data
- Exempted from the curse of dimensionality

Deep Learning for Solving Forward and Inverse Problems

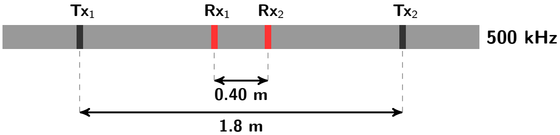
Governing PDEs in Electromagnetism



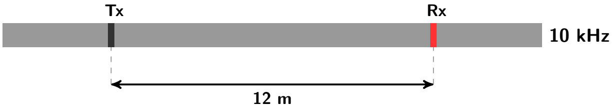
Maxwell's equations

$$
\begin{cases}
\nabla \times \mathbf{H} &= (\boldsymbol{\sigma} + j\omega\boldsymbol{\epsilon})\mathbf{E} + \mathbf{J}^{imp} & \text{Ampère's law,} \\
\nabla \times \mathbf{E} &= -j\omega\boldsymbol{\mu}\mathbf{H} + \mathbf{M}^{imp} & \text{Faraday's law,} \\
\nabla \cdot (\boldsymbol{\epsilon}\mathbf{E}) &= \rho_e & \text{Gauss' law of electricity,} \\
\nabla \cdot (\boldsymbol{\mu}\mathbf{H}) &= 0 & \text{Gauss' law of magnetism.}
\end{cases}
$$

Borehole Synthetic Example: Input Measurements

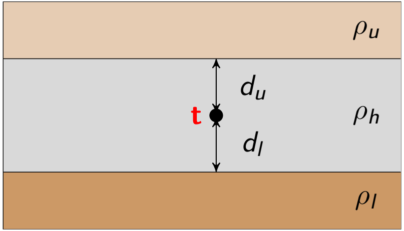


- Co-axial attenuation and phase difference



- Co-axial attenuation and phase difference
- Geosignal

Geophysics Synthetic Example: Output Earth Parametrization



$\rho_u \in [1, 10^3]\Omega \cdot m$: Upper layer resistivity

$\rho_h \in [1, 10^3]\Omega \cdot m$: Central layer resistivity

$\rho_l \in [1, 10^3]\Omega \cdot m$: Lower layer resistivity

$d_u \in [10^{-2}, 10]m$: Vertical distance to upper layer

$d_l \in [10^{-2}, 10]m$: Vertical distance to lower layer

Geophysics Synthetic Example: Loss Function

Definitions:

$$\begin{aligned}
\mathcal{F} &:= \text{Forward operator (Earth properties} -- > \text{Measurements)} \\
\mathcal{I} &:= \text{Inverse operator (Measurements} -- > \text{Earth properties)} \\
\mathcal{I}_{\phi^*} &:= \text{Neural Network approx. of } \mathcal{I}
\end{aligned}$$

Desired loss function:

$$\mathcal{I}_{\phi^*} := \arg \min_{\mathcal{I}_\phi, \phi \in \Phi} \sum_i \|(\mathcal{F} \circ \mathcal{I}_\phi)(\mathbf{m}_i) - \mathbf{m}_i\|$$

Geophysics Synthetic Example: Loss Function

Definitions:

$$
\begin{aligned}
\mathcal{F} \quad &:= \text{Forward operator (Earth properties} \; -- > \text{Measurements)} \\
\mathcal{I} \quad &:= \text{Inverse operator (Measurements} \; -- > \text{Earth properties)} \\
\mathcal{I}_{\phi^*} \quad &:= \text{Neural Network approx. of } \mathcal{I}
\end{aligned}
$$

Desired loss function:

$$
\mathcal{I}_{\phi^*} \;:= \arg \min_{\mathcal{I}_\phi, \phi \in \Phi} \sum_i \|(\mathcal{F} \circ \mathcal{I}_\phi)(\mathbf{m}_i) - \mathbf{m}_i\|
$$

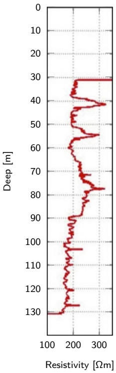
We approximate the full inverse function $\mathcal{I}$

Deep Neural Networks (Deep Learning) for Inverse Problems

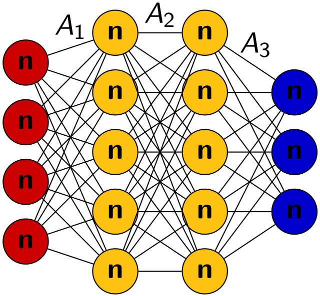
$$\text{Approximate: } \mathcal{I} \approx \mathcal{I}_\phi := A_k \circ N \circ A_{k-1} \circ \cdots \circ N \circ A_1$$

N – Non-linear activation function ; A_k – Affine transformation

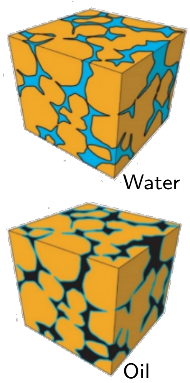
Input



Fully connected network



Output

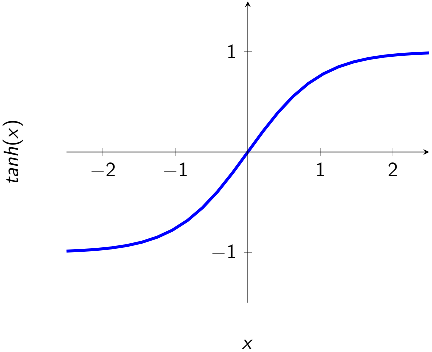


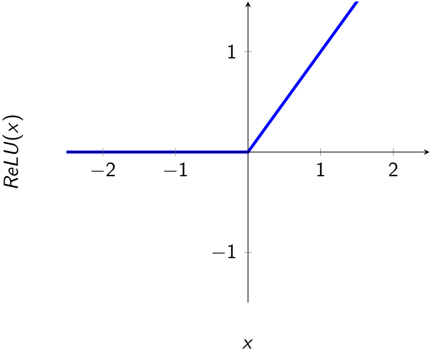
Deep Neural Networks (Deep Learning) for Inverse Problems

Approximate: $\mathcal{I} \approx \mathcal{I}_\phi := A_k \circ N \circ A_{k-1} \circ \cdots \circ N \circ A_1$

A_k – Affine transformation: $A_k \cdot x + b_k$

N – Non-linear activation function:





Geophysics Synthetic Example: Loss Function

Definitions:

$$\mathcal{F} := \text{Forward operator (Earth properties} -- > \text{Measurements)}$$

$$\mathcal{I} := \text{Inverse problem (Measurements} -- > \text{Earth properties)}$$

$$\mathcal{I}_{\phi^*} := \text{Neural Network approx. of } \mathcal{I}$$

$$\mathcal{I}_{\phi^*} := \arg \min_{\mathcal{I}_\phi, \phi \in \Phi} \sum_i \|(\mathcal{F} \circ \mathcal{I}_\phi)(\mathbf{m}_i) - \mathbf{m}_i\|$$

Evaluating $\mathcal{F}$ is expensive

Geophysics Synthetic Example: Loss Function

Definitions:

$$\begin{aligned} \mathcal{F} \quad &:= \text{Forward operator (Earth properties} -- > \text{Measurements)} \\ \mathcal{I} \quad &:= \text{Inverse problem (Measurements} -- > \text{Earth properties)} \\ \mathcal{I}_{\phi^*} &:= \text{Neural Network approx. of } \mathcal{I} \\ \mathcal{F}_{\theta^*} &:= \text{Neural Network approx. of } \mathcal{F} \end{aligned}$$

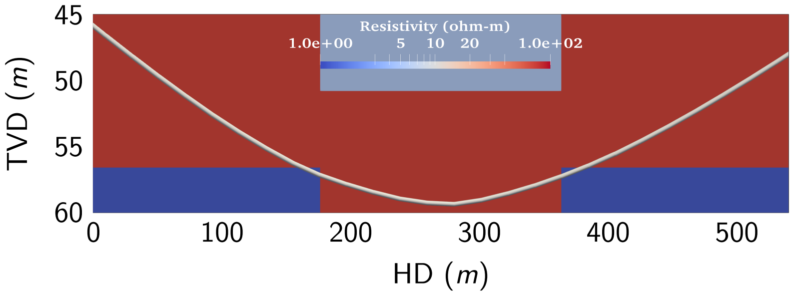
Two-step based loss function:

$$\mathcal{F}_{\theta^*} := \arg\min_{\theta\in\Theta} \sum_i \|\mathcal{F}_\theta(\mathbf{z}_i) - \mathcal{F}(\mathbf{z}_i)\|$$

$$\mathcal{I}_{\phi^*} := \arg\min_{\phi\in\Phi} \sum_i \|(\mathcal{F}_{\theta^*} \circ \mathcal{I}_\phi)(\mathbf{m}_i) - \mathbf{m}_i\|$$

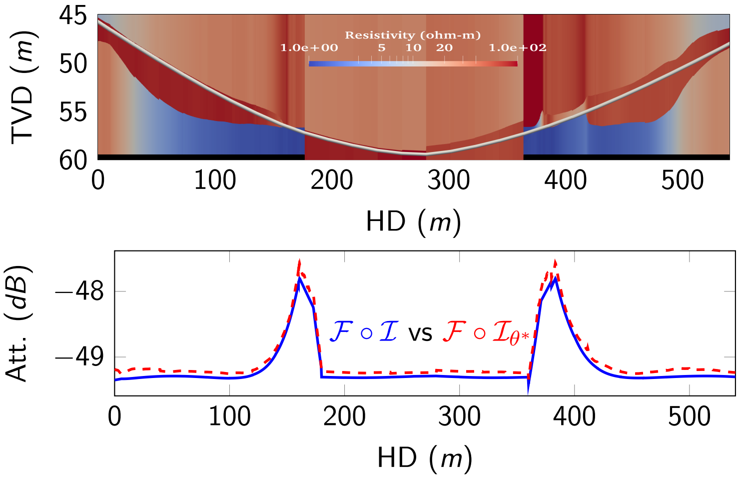
Shahriari, M., Pardo, D., Rivera, J. A., Torres-Verdín, C., Picon, A., Del Ser, J., Ossandón, S., Calo, V. M.: Error control and loss functions for the deep learning inversion of borehole resistivity measurements. International Journal for Numerical Methods in Engineering **122**(6), 1629–1657 (2021)

Synthetic Example

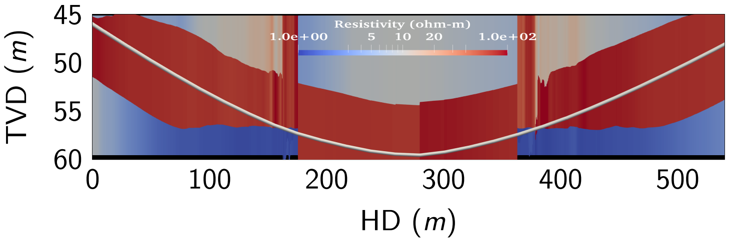


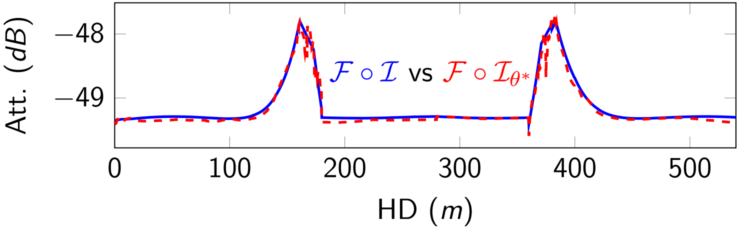
Formation of synthetic example.

Numerical Results: Two Step based



Numerical Results with Regularization

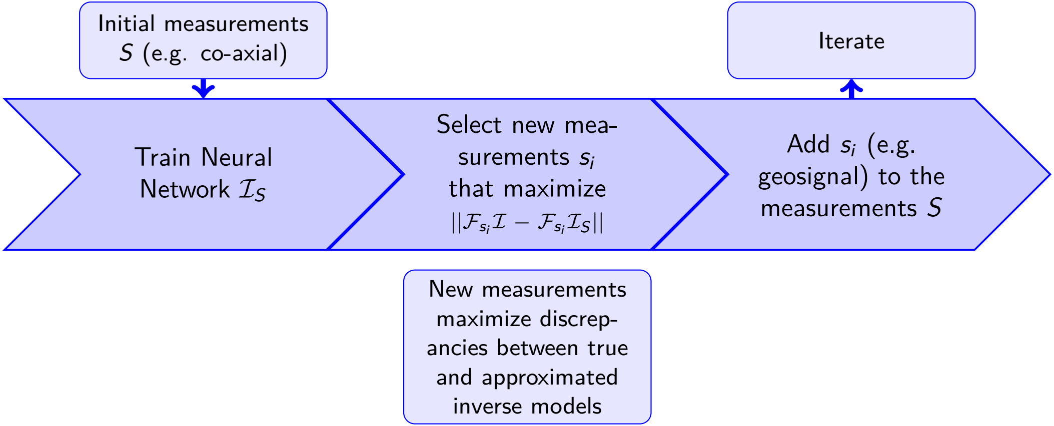




Optimization of measurement system

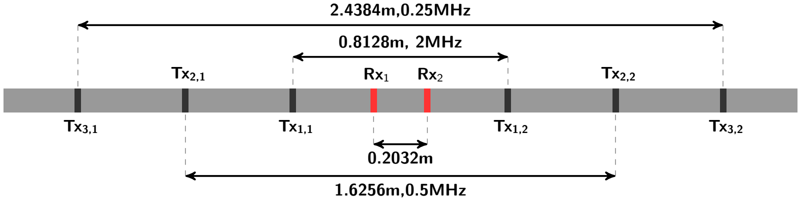
Design of Measurement Acquisition System

We select measurements following this iterative algorithm:



Measurement Tool

- Conventional Logging-while-Drilling (LWD)

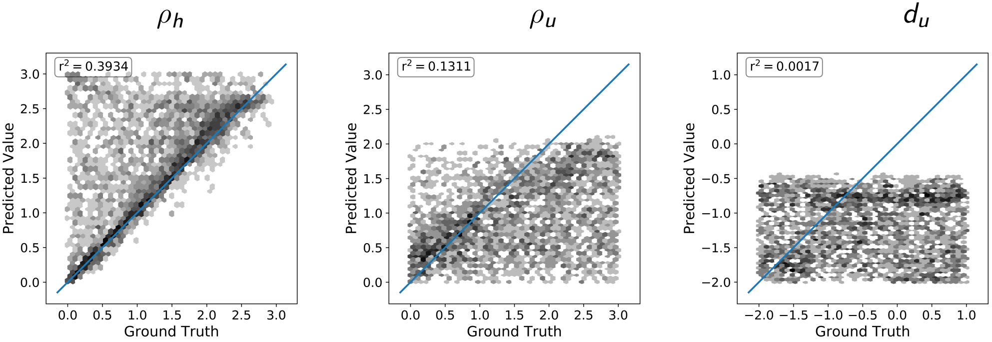


- Deep Azimuthal

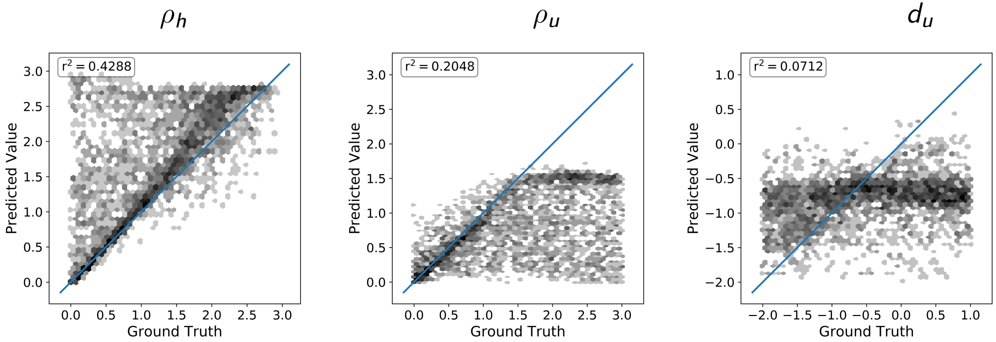
Types of Measurements

Name	Measured Component	LWD	Deep Azimuthal
zz	H_{zz}	✓	✓
xx	H_{xx}	✓	✓
yy	H_{yy}	✓	✓
xxyyzz+	$H_{xx} + H_{yy} + H_{zz}$	✓	✓
Geosignal	$\frac{H_{zz} - H_{zx}}{H_{zz} + H_{zx}}$	✓	✓
Symmetrized directional	$\frac{H_{zz} + H_{zx}}{H_{zz} - H_{zx}} \cdot \frac{H_{zz} - H_{xz}}{H_{zz} + H_{xz}}$	✓	✓
Antisymmetrized directional	$\frac{H_{zz} + H_{zx}}{H_{zz} - H_{zx}} \cdot \frac{H_{zz} + H_{xz}}{H_{zz} - H_{xz}}$	✓	✓
Harmonic resistivity	$\frac{H_{xx} + H_{yy}}{2\ H_{zz}}$	✓	✓
Harmonic anisotropy	$\frac{H_{xx}}{H_{yy}}$	✓	✓

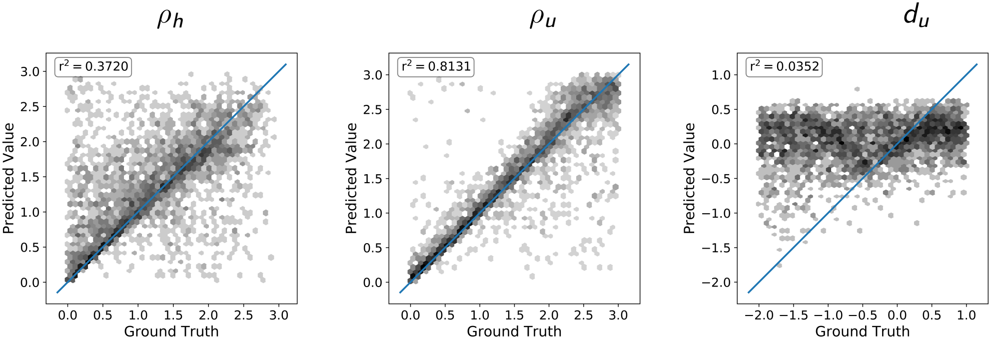
Iter 0: 30k Samples



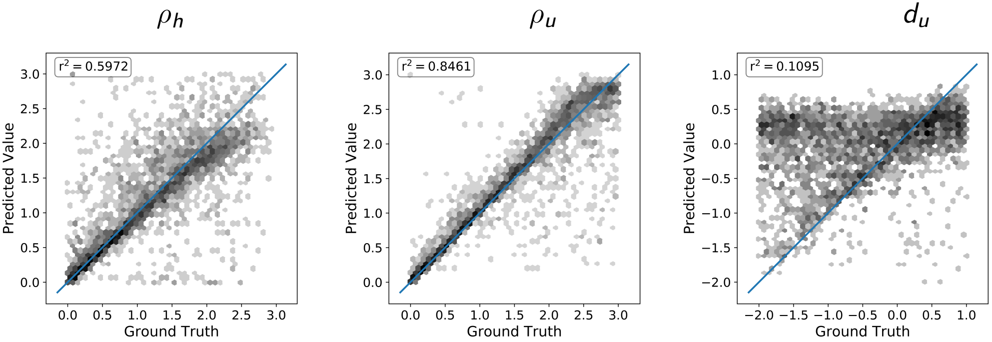
Iter 1: 30k Samples



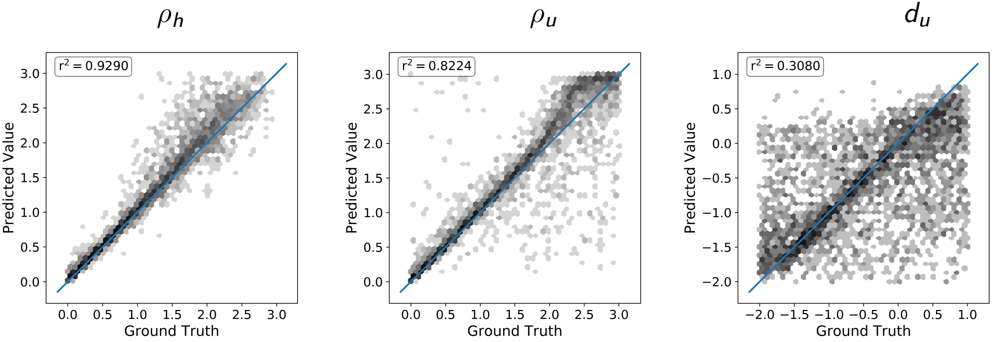
Iter 2: 30k Samples



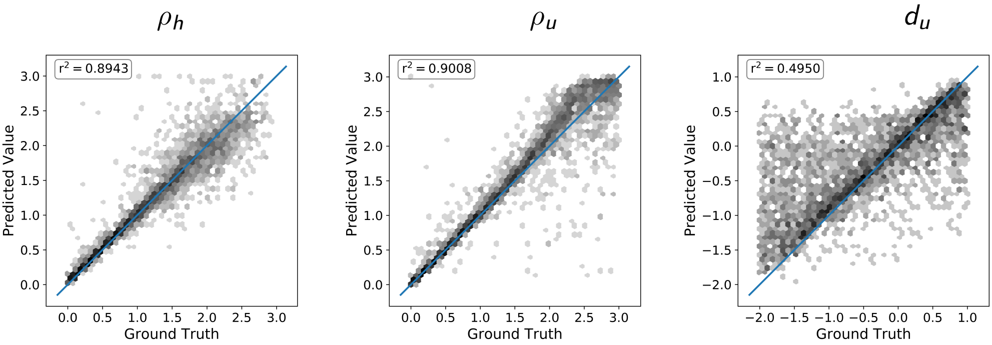
Iter 3: 30k Samples



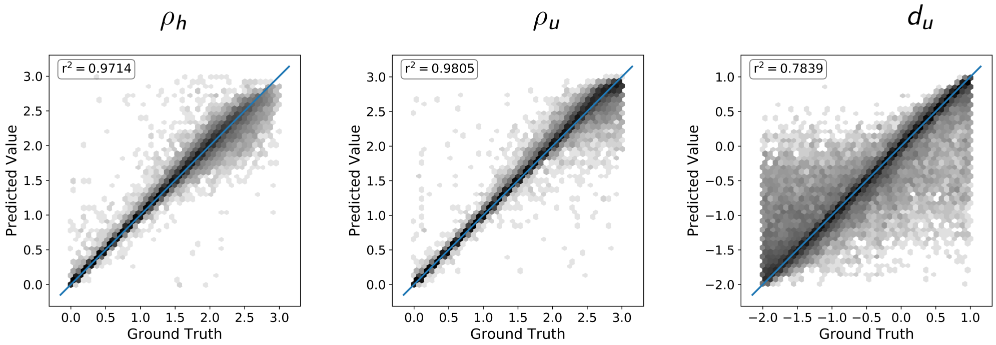
Iter 4: 30k Samples



Iter 5: 30k Samples

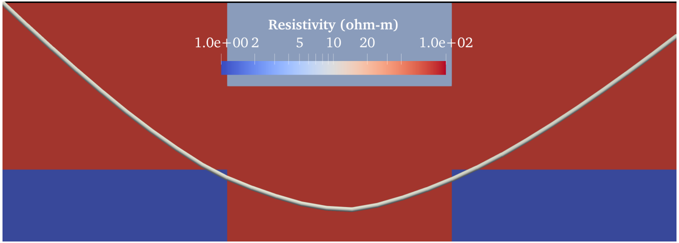


Iter 5: 300k Samples

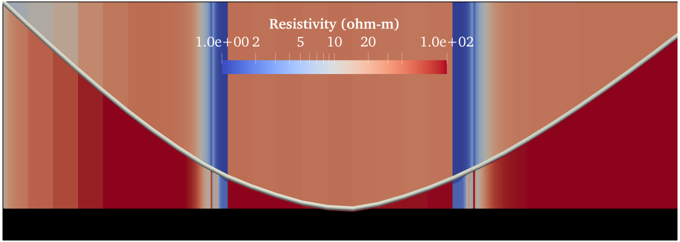


Iter 0: 30k Samples

Original:

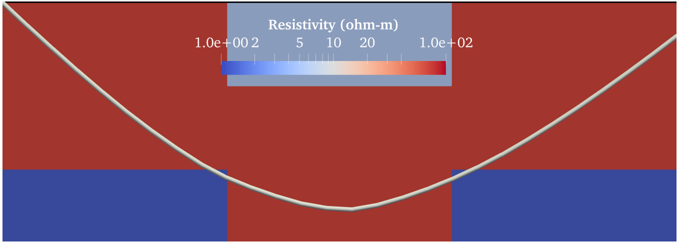


Inverted:

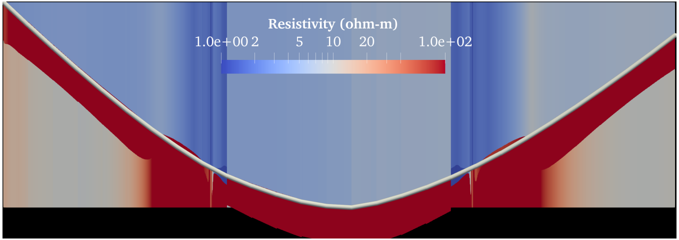


Iter 1: 30k Samples

Original:

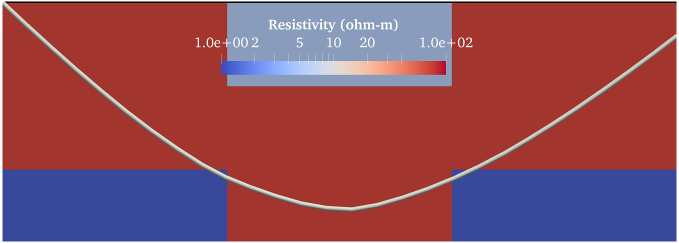


Inverted:

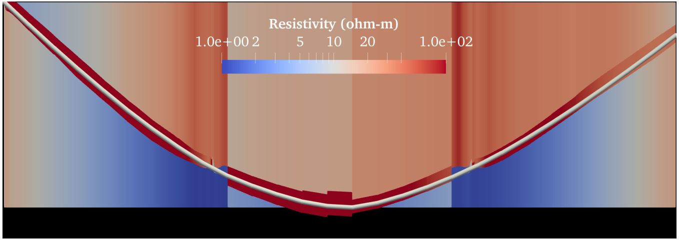


Iter 2: 30k Samples

Original:

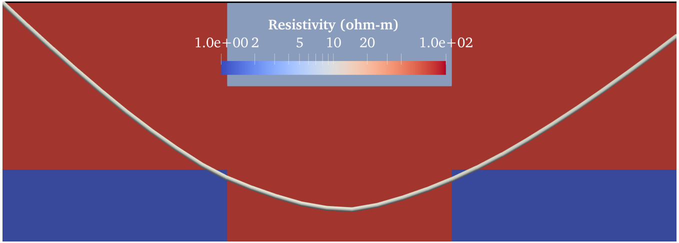


Inverted:

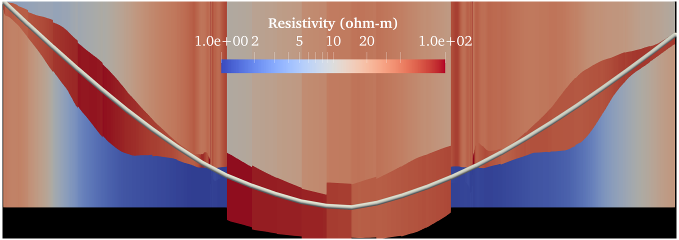


Iter 3: 30k Samples

Original:

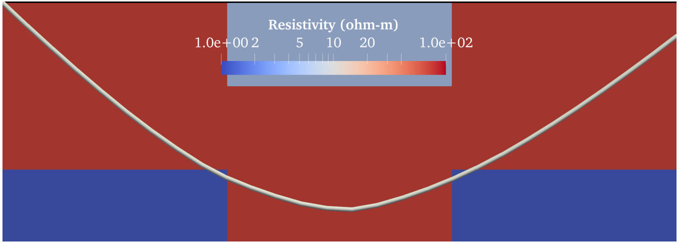


Inverted:

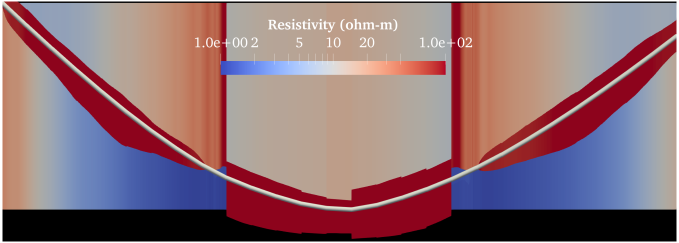


Iter 4: 30k Samples

Original:

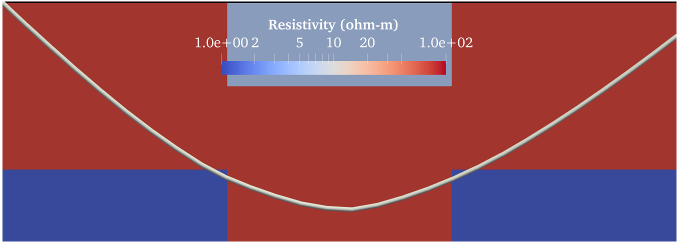


Inverted:

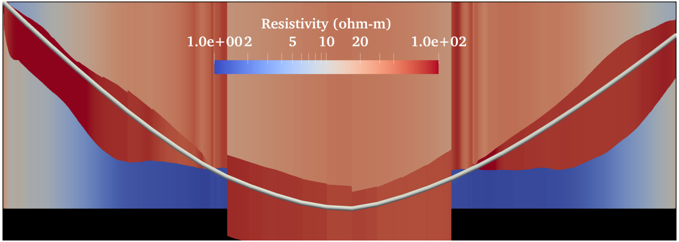


Iter 5: 30k Samples

Original:

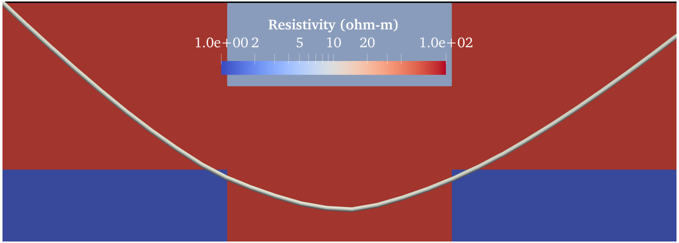


Inverted:

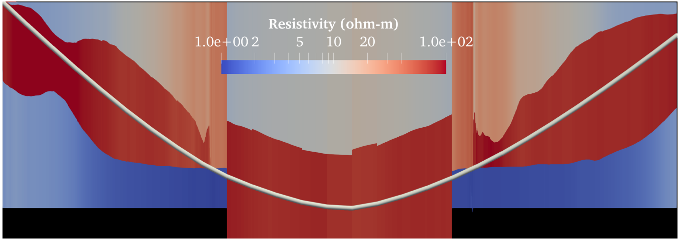


Iter 5: 300k Samples

Original:



Inverted:



Database generation

Motivation and Objectives

- The inversion process requires a massive database that relates multiple Earth models to borehole resistivity measurements.
- We often produce an *offline* synthetic database using tens of thousands of simulations by solving the Maxwell's equations with different Earth models.
- The objective is to efficiently generate a massive database for 2.5D borehole resistivity measurements
- We employ **refined isogeometric analysis (rIGA)** as a high-performance computational method to perform rapid and accurate simulations.

Auto ML

Why AutoML?

- Architecture design by hand is complex.

- It is possible to use a large DNN to achieve high accuracy:
  - It imposes unnecessary high computational costs while training.
  - It may cause overfitting.
  - It requires high memory and processor capabilities during evaluation.

Goal

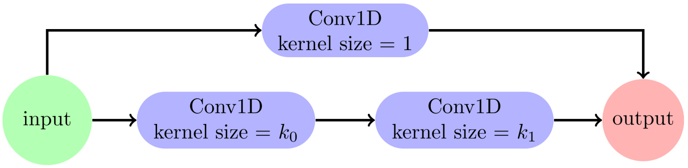
To find a DNN architecture that delivers an acceptable level of accuracy with a minimum number of parameters.

Hyperparameter tuning

- Input
  - Search space of hyperparameters (DNN architectures)
  - Dataset
  - Scoring function, e.g., loss function
  - Stopping criteria, if needed

- Output
  - The optimal hyperparameters (DNN architecture) corresponding to the dataset

Search space



- n : number of blocks
- k_0, k_1: kernel sizes of the convolutional layers

- Forward model: $S_{\mathcal{F}} = \{n = \{1, 2, 3, 4\}; k_0, k_1, L = \{3, 5, 7\}\}$
  - L: the kernel size of a final convolutional output layer
- Inverse model: $S_{\mathcal{I}} = \{n = \{1, 2, 3, 4, 5\}; k_0, k_1 = \{3, 5, 7\}\}$
  - The output layer is dense

Scoring function

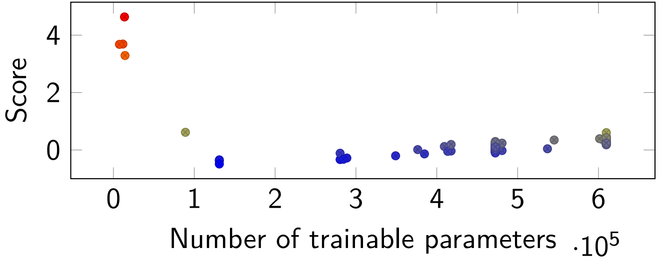
$$R(h) = \underbrace{\frac{\mathcal{H}(h) - \mathcal{H}(h^o)}{\mathcal{H}(h^o)}}_{\text{relative error}} - \underbrace{\frac{N_p(h^o) - N_p(h)}{N_p(h^o)}}_{\substack{\text{relative increase in} \\ \text{the number of unknowns}}},$$

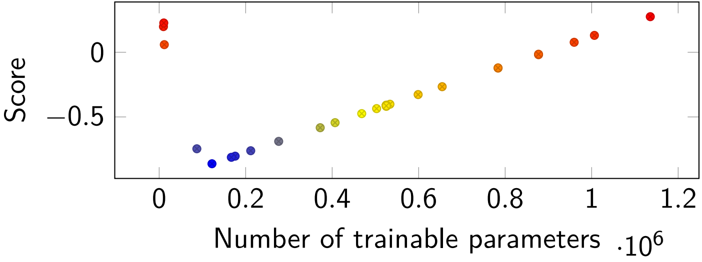
- h^o: Hyperparameters of a reference model
- N_p: Number of unknowns
- Forward operator: $\mathcal{H}(h_f) = \sum_{i=1}^{n_v} \|\mathcal{F}_{h_f,\alpha^*}(t_i, \mathbf{p}_i) - \mathbf{m}_i\|$
- Inverse operator: $\mathcal{H}(h_i) = \sum_{i=1}^{n_v} \|\mathcal{F}_{h_f^*,\alpha^*} \circ \mathcal{I}_{h_i,\beta^*}(t_i, \mathbf{m}_i) - \mathbf{m}_i\|$
- n_v: Number of validation samples

Tunning optimization problem

$$h^* = \underset{h \in \mathcal{S}}{\arg\min}\, R(h),$$

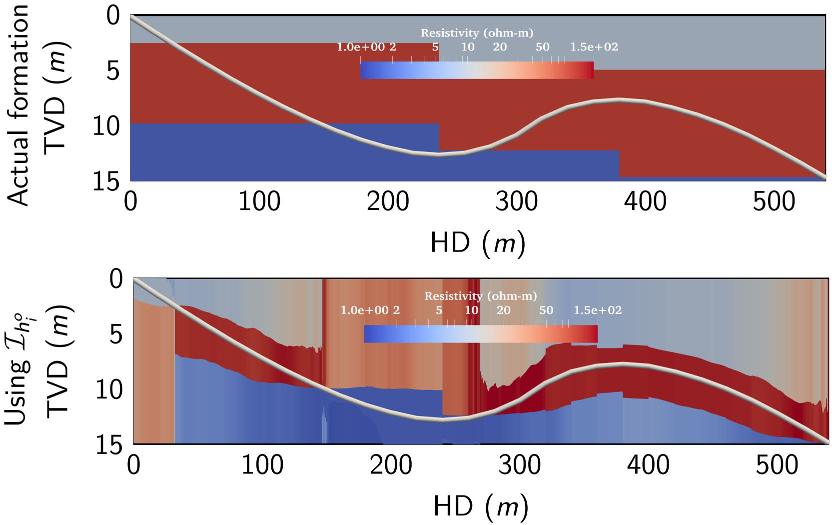
Tuning results: Bayesian optimization



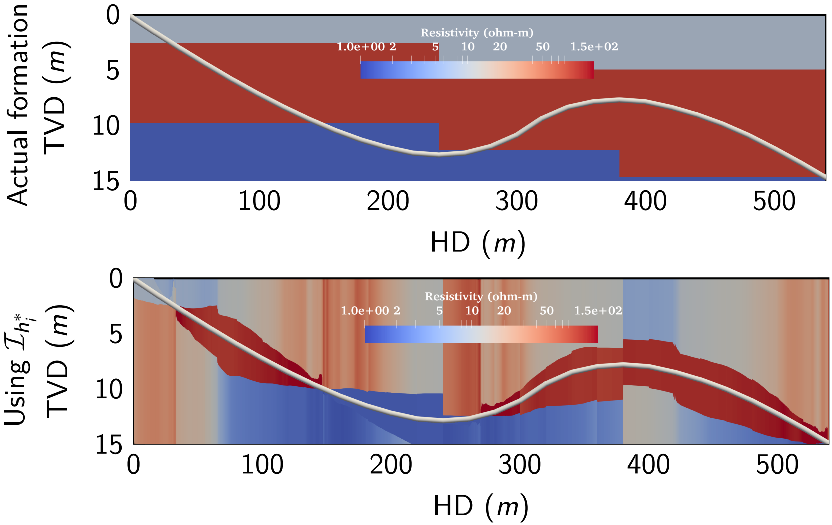


- Forward:
  - Original DNN: 525k parameters
  - Optimal DNN: 131k parameters
- Inverse:
  - Original DNN: 890k parameters
  - Optimal DNN: 122k parameters

Inversion results: using original DNN



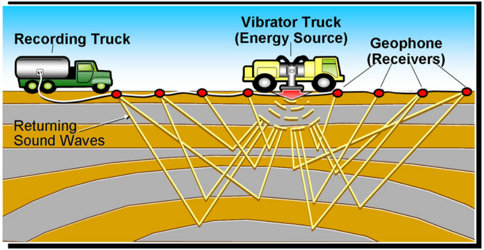
Inversion results: using optimal DNN



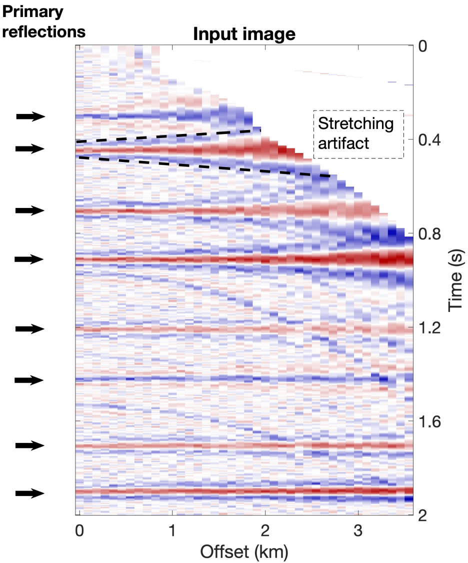
Application on Seismic

Surface seismic measurements

- In seismic studies, mechanical waves are generated by a source and recorded with several receivers.
- Different processing techniques are required to produce 2D or 3D images of subsurface properties.
- Postprocessing requires high computational and/or user costs that can be alleviated using machine learning.



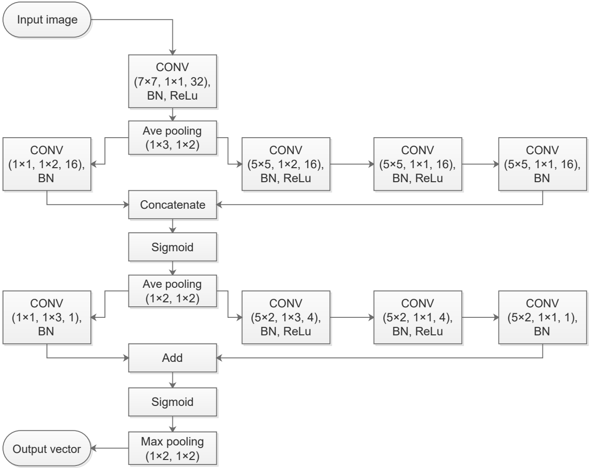
Seismic Application 1



- *Wavelet stretching* is an important artifact in conventional data processing.
- To rectify it, we propose a method that needs to recognize primary reflected signals in partially processed recorded data.
- We use deep learning to recongnize the primaries.

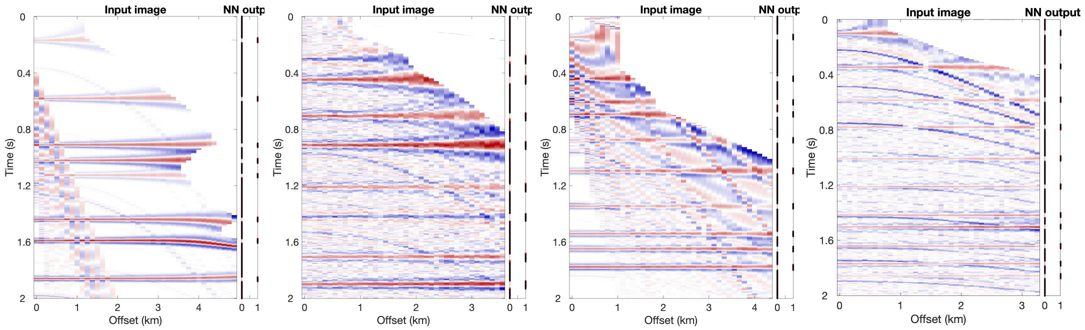
Seismic Application 1

- We design a ResNet architecture and generate 40,000 synthetic training data samples.



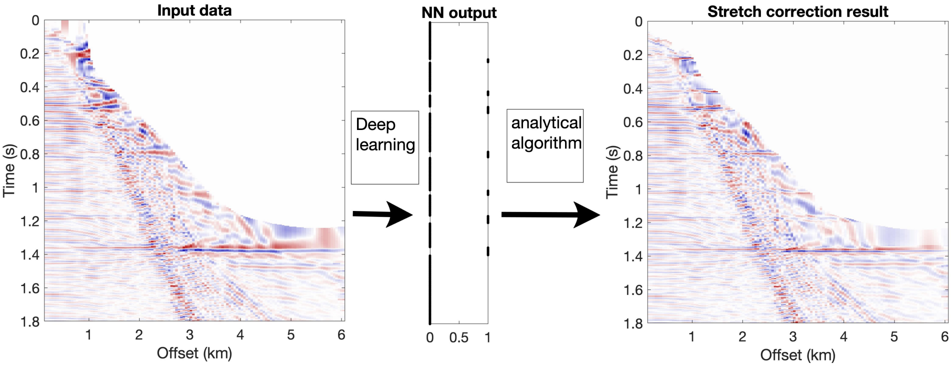
Seismic Application 1

- Examples of the training data samples, where the input image and the output vector are shown:



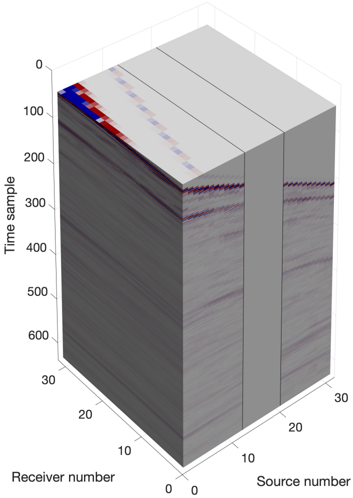
Seismic Application 1

- The DNN output is used in our analytical artifact correction algorithm to make it fully automatic.



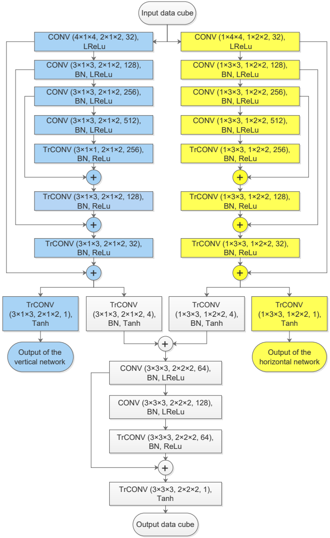
Abedi, M. M., and Pardo, D. (2022). Nonhyperbolic normal moveout stretch correction with deep learning automation. Geophysics, 87(2), U57-U66..

Seismic Application 2



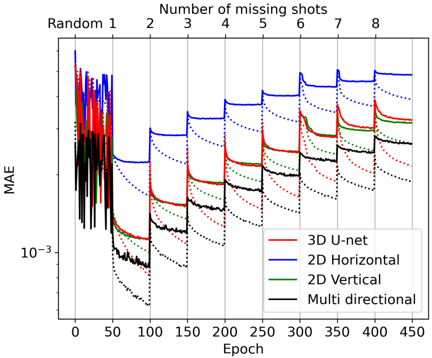
- We observe gaps in the recorded data due to difficulties in deployment of several sources or receivers.
- We propose a new self-supervised method called "multidirectional deep learning" to fill these gaps (extrapolation).

Seismic Application 2



- We mix two 2D networks (corresponding to horizontal and vertical slices of data) in a 3D network.

Seismic Application 2



- The proposed method is more accurate than a conventional 3D U-net.

Missing shots	1	2	3	4	5	6	7	8
3D U-net	1.1	1.5	1.8	2.1	2.3	2.6	2.9	3.1
2D vertical	1.1	1.5	1.8	2.0	2.3	2.5	2.7	2.9
2D horizontal	1.2	1.5	1.8	2.0	2.3	2.5	2.6	2.8
Multidirectional	**0.9**	**1.1**	**1.3**	**1.6**	**1.7**	**1.9**	**2.0**	**2.2**

Mean absolute error of the test synthetic data. The values should be multiplied by 10^{-3}

Conclusions and Future Work

Conclusions

- Deep Learning (DL) is a promising alternative for solving geophysical inverse problems.
- DL opens the alternative for solving challenging geophysical problems that could not be solved with traditional methods.
- We need efficient DL solvers of Partial Differential Equations (PDEs).

Call for Ph.D. Students and Postdoctoral Fellows

Ph.D. & postdoctoral fellowships in Bilbao

Nice place...

...good maths!

Contact: dzubiaur@gmail.com